Team Ade to Order -- Adrian Kloskowski, Kashf Mashrafi, Ying Xin Jiang

APCS2 pd8

Project Proposal

2018-05-29

2D Physics Engine: Angry Birds

Our final project aims to create an engine in Processing to simulate the movement and interaction of rigid bodies within a two-dimensional space. Using vector physics, our engine will be able to generate simplistic shapes (polygons and circles) and use vector physics and collisions to simulate their movement, taking into account gravity, friction, and mass.

As a simple & practical demonstration of the physics engine, we will create a copy of the Angry Birds game. The player will be able to direct projectiles (the aforementioned birds, which happen to be angry) towards pre-built towers of polygons resting atop each other (containing circular targets, the pigs). The goal of this simple challenge is to use the avian projectiles to topple the tower.

What It Uses:

- Processing - For visual display and coding.
- Object-oriented programming and class hierarchies - Used to classify, keep track of, and control objects on-screen as well as their properties and functions.
    - SubClasses - for seperate variable objects (e.g. circles & rectangles) with needed properties.
- Vector physics formulae - For realistic portrayal of physics mechanics in processing.
- Data collections - Used to properly assess and control objects in engine.
    - Array Priority Queue - To control the modifications and updates of objects per frame based on specific properties.

Priorities:

1. Create rigid bodies (rectangles and circles) that are affected by gravity and do not intersect with one another (impulse resolution).
2. Implement the even distribution of velocities when multiple rigid bodies collide.
3. Implement a way for a user to generate, edit, delete, and otherwise interact with shapes within an abstract debug environment.
4. Implement rotational physics.
5. Implement friction along surfaces.
6. Start work on Angry Birds game, beginning with player interaction (shooting the slingshot).
7. Create levels (wooden towers with pigs) for the player to topple.

8. Implement victory condition, as well as limitation on attempts before level is reset.
9. If possible, add polish: add images in place of abstract shapes, pretty up GUI, etc.
10. If possible, add multiple levels & a level select.